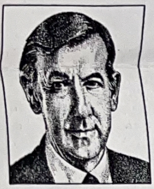
APPLIED INTELLIGENCE

AD/Cycle Promises Mix-and-Match Development Tools



JAMES MARTIN

This is the sixth in a series of articles covering the architectures, methodologies and tools required to retool information systems.

With the spectrum of incompatible computer-aided software engineering (CASE) tools currently topping the 200 mark, it's nearly impossible to share design information among diverse products. To overcome these limitations and allow developers to mix and match assorted tools, it's essential that CASE vendors subscribe to a common standard for design information.

CASE tools accept design information in the form of diagrams, analyze the information for logical errors and then store the design information in a data dictionary or repository. Today, however, each vendor stores information within its own repository differently; consequently, there are hundreds of ways to represent data-flow diagrams.

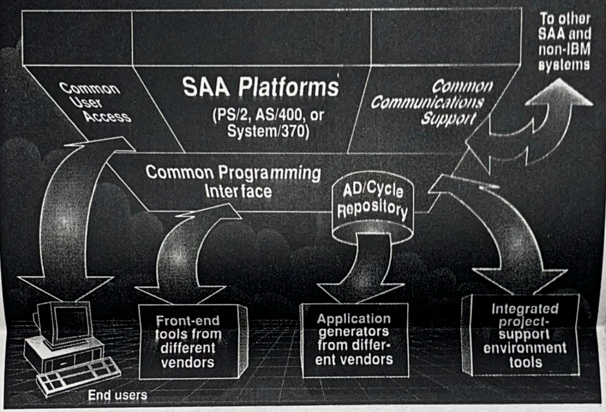
This situation is about to change as a result of IBM's announcement last September of its AD/Cycle repository, which specifies a standard way of defining design information in the form of a common meta-data model. The meta-data model offers a standard representation for the information contained in data-flow, decomposition, entity-relationship and other diagrams.

To date, 47 vendors have announced that they will reimplement their CASE repositories for AD/Cycle compliance. Once this happens, developers will be presented with a "software backplane" environment, which will let CASE tools from multiple vendors easily share design information.

This software-backplane environment is illustrated in the accompanying figure: A variety of front-end CASE tools, back-end code generators and support tools all share information via the common AD/Cycle repository standard.

AD/Cycle Application-Development Environment

Objective Is To Let Developers Take Advantage of SAA-Compliant CASE Tools from Different Vendors; All Tools Share a Common Repository


To other SAA and non-IBM systems
Common User Access
SAA Platforms
(PS/2, AS/400, or System/370)
Common Communications Support
Common Programming Interface
AD/Cycle Repository
Front-end tools from different vendors
Application generators from different vendors
Integrated project-support environment tools
End users

John Avakian

Front-end tools, which feed data into the repository, could include planning, analysis and design workbenches from a variety of CASE vendors. Back-end tools, which extract specification information from the repository to perform specific tasks, could consist of source-code generators for COBOL, C and Ada, or modules that generate executable load modules.

Compliance with a common repository of design information, as defined by AD/Cycle, will permit organizations to select CASE tools that are the "best of the breed," or tools that best support a specific portion of the development life cycle. Eventually, the hope is that users will mix and match tools from multiple vendors to support specific design and implementation requirements.

To this end, an organization might select one CASE tool for database design, another to support strategic planning tasks, and yet another to handle analysis and design operations. In addition, they could choose from a wide range of code generators to produce applications quickly in C, COBOL, Ada and other source languages.

Unfortunately, this era of "plug-and-play" CASE tools is still off in the future. While most important CASE tool vendors have pledged compatibility with AD/Cycle, these products are still not available and are not expected to surface until late this year and throughout 1991.

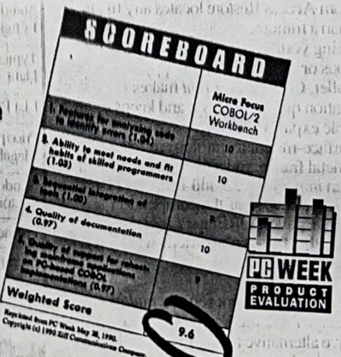
To support the entire applications-development life cycle today, an organization must purchase an integrated set of CASE tools from a single vendor. However, there are only a handful of vendors selling integrated CASE tools.

Once products appear that conform to the AD/Cycle repository standard and are compatible with the services provided by IBM's Systems Application Architecture (SAA), organizations can begin developing true cooperative-processing applications. This type of application will support the distribution of both process and data throughout a network of computers.

As shown in the figure, the AD/Cycle environment provides a set of SAA services, including a common user interface, a common programming interface, a common communications interface and common repository services.

Next week, I will discuss the features users should look for in a CASE tool to obtain the maximum return from an investment in CASE technology. ■

The concepts embodied in this article are described in the CASE volume in The James Martin Report Series. For more information on this volume, call (617) 639-1958. For information on seminars, contact (in the United States and Canada) Technology Transfer Institute, 741 10th St., Santa Monica, Calif. 90402 (213) 394-8305. In Europe, contact Savant, 2 New St., Carnforth, Lancs., LA5 9BX United Kingdom (0524) 734 505.



SCOREBOARD

	Micro Focus COBOL/2 Workbench
1. Features for analyzing code to identify errors (1.04)	10
2. Ability to meet needs and fit habits of skilled programmers (1.03)	10
3. Sequential integration of tools (1.00)	9
4. Quality of documentation (0.97)	10
5. Quality of support for rehosting mainframe applications on PC-based COBOL implementations (0.97)	9
Weighted Score	9.6

Reprinted from PC Week May 28, 1990.
Copyright (c) 1990 Ziff Communications Company.